AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for caching in a tracing framework, comprising:
   firing a probe associated with a thread;
   evaluating a first predicate of the probe to determine a first Boolean value, wherein the first Boolean value is false;
   determining that the first predicate is cacheable;
   identifying the first predicate using a predicate cache identifier;
   storing the predicate cache identifier with the probe as a probe cache identifier;
   caching the ~~first predicate~~ first Boolean value and the predicate cache identifier in a predicate cache associated with the thread, based on the first Boolean value being false and the first predicate being cacheable ~~the evaluating of the first predicate and cacheability of the first predicate~~; and
   transferring control to the thread, based on the caching.

2. (Cancelled)

3. (Cancelled)

4. (Currently Amended) The method according to claim [[3]] 1, further comprising:
   executing an action of the probe when the Boolean value is true.

5. (Cancelled)

6. (Cancelled)

7. (Currently Amended) The method according to claim [[6]] 1, wherein cacheable is the first predicate referencing an immutable variable.

8. (Currently Amended) The method according to claim [[6]] 1, wherein cacheable is the first predicate referencing a thread-specific variable.

9. (Cancelled)

10. (Currently Amended) The method according to claim 1, wherein the transferring occurs if the ~~first predicate~~ first Boolean value is cached in the predicate cache.

11. (Currently Amended) The method according to claim [[9]] 1, wherein the probe further encounters a second predicate of the probe.

12. (Original) The method according to claim 11, further comprising:
evaluating the second predicate of the probe.

13. (Currently Amended) The method according to claim 12, wherein [[the]] evaluating the second predicate comprises determining a second Boolean value of the second predicate.

14. (Currently Amended) The method according to claim 13, wherein the second Boolean value is true.

15. (Original) The method according to claim 14, further comprising:
executing an action of the probe.

16. (Currently Amended) The method according to claim 13, wherein the second Boolean value is false.

17. (Original) The method according to claim 16, further comprising:
determining whether the second predicate is cacheable.

18. (Previously Presented) The method according to claim 17, further comprising:
identifying the second predicate using the predicate cache identifier, if the first predicate and the second predicate are the same.

19. (Original) The method according to claim 18, wherein cacheable is the first predicate referencing an immutable variable and the first predicate and the second predicate having the same identifier.

20. (Original) The method according to claim 18, wherein cacheable is the first predicate referencing a thread-specific variable and the first predicate and the second predicate having the same identifier.

21. (Currently Amended) The method according to claim 1, further comprising:
determining whether the ~~first predicate~~ first Boolean value is cached; and
determining whether the predicate cache is valid.

22. (Currently Amended) The method according to claim 21, wherein the determining whether the ~~predicate~~ first Boolean value is cached comprises comparing whether a probe cache identifier and a predicate cache identifier stored in the predicate cache are equivalent.

23. (Previously Presented) The method according to claim 21, wherein determining whether the predicate cache is valid comprises comparing whether a probe cache identifier and a predicate cache identifier stored in the predicate cache are non-zero.

24. (Original) The method according to claim 1, further comprising:
invalidating the predicate cache.

25. (Original) The method according to claim 24, wherein the invalidating comprises setting the predicate cache to zero.

26. (Original) The method according to claim 24, wherein the invalidating is a result of a thread-specific variable being stored.

27. (Original) The method according to claim 1, further comprising:
setting the predicate cache to zero initially.

28. (Currently Amended) A computer system for caching in a tracing framework comprising:
a processor;
a memory;
a storage device; and
software instructions stored in memory for enabling the computer system to:
fire a probe associated with a thread;
evaluate a predicate of the probe to determine a Boolean value, wherein the Boolean value is false;
determine that the predicate is cacheable;
identify the predicate using a predicate cache identifier;
store the predicate cache identifier with the probe as a probe cache identifier;

cache ~~the predicate~~ Boolean value and the predicate cache identifier in a predicate cache associated with the thread, based on the Boolean value being false and the predicate being cacheable ~~the evaluating of the predicate and cacheability of the predicate~~; and

transfer control to the thread, based on the caching.

29. (New) The method of claim 1, wherein false means an integer value of zero.

30. (New) The method of claim 1, wherein false means an integer value of one.

31. (New) The system of claim 28, wherein false means an integer value of zero.

32. (New) The system of claim 28, wherein false means an integer value of one.

33. (New) The method of claim 1, wherein the predicate cache identifier is a thirty-two bit, non-zero integer.